

## ⑫ 公開特許公報(A)

昭62-140554

⑤Int.Cl.<sup>4</sup>

識別記号

庁内整理番号

④公開 昭和62年(1987)6月24日

H 04 N 1/415

8220-5C

審査請求 未請求 発明の数 1 (全6頁)

⑬発明の名称 デジタル画像の符号化方式

⑭特 願 昭60-281641

⑮出 願 昭60(1985)12月13日

⑯発明者	鈴木	良行	東京都大田区下丸子3丁目30番2号	キャノン株式会社内
⑯発明者	船田	正広	東京都大田区下丸子3丁目30番2号	キャノン株式会社内
⑯発明者	佐藤	衛	東京都大田区下丸子3丁目30番2号	キャノン株式会社内
⑯発明者	佐藤	幸夫	東京都大田区下丸子3丁目30番2号	キャノン株式会社内
⑰出願人	キャノン株式会社			東京都大田区下丸子3丁目30番2号
⑱代理人	弁理士 丸島 儀一			

## 明 細 書

## 1. 発明の名称

デジタル画像の符号化方式

## 2. 特許請求の範囲

- (1) デジタル画像データを  $N \times N$  のブロック単位に分割し、ブロックのパターンを表すコードとそのパターンの状態を表すコードとにより各ブロックを符号化することを特徴とするデジタル画像の符号化方式。
- (2) 上記状態はパターンの回転および対称のモードであることを特徴とする特許請求の範囲第(1)項に記載のデジタル画像の符号化方式。
- (3) 回転および対称を表すコードに処理を加えることで、その符号が表わすパターンを回転および対称処理を施したパターンの符号とすることを特徴とする特許請求の範囲第(2)項に記載のデジタル画像の符号化方式。

## 3. 発明の詳細な説明

〔技術分野〕

本発明は、ブロック単位で符号化するデジ

タル画像の符号化方式に関するものである。

〔従来技術〕

デジタル画像を所定サイズのブロックに分割し、そのブロックの画像パターンをブロック単位で符号化する方法があるが、この方式で符号化すると、画像を符号化したままの状態と比較的容易に画像の移動・回転などの画像処理をすることができる。それは、この方式ではブロック単位で符号化してあるので、他のブロックに影響を与えずにそのブロックだけ処理が独立して行なえるからである。

しかし、ここで気を付けなければならないことは、回転の処理をする場合、あくまでもブロック単位で処理がされただけで、このまま復号化してもその再生像は完全でないことである。

これを第6図を使って説明する。第6図で0がオリジナルの画像とすると、これをブロック単位で符号化して、時計方向に90度回転する例を示している。まず、オリジナル画像0は4×4のブロックサイズの4つのブロック01～04

に分割される。そして、それぞれのブロック  
01~04のパターンを符号器11によつて $B_{11}$ 、 $B_{12}$ 、 $B_{21}$ 、 $B_{22}$ に符号化し、メモリ12に記憶する。  
ここで、時計方向に90°回転するためには、回転処理がない場合におけるメモリ12からの読出しを $B_{11}$ 、 $B_{12}$ 、 $B_{21}$ 、 $B_{22}$ という順番だとすれば、第6図に示したように $B_{21}$ 、 $B_{11}$ 、 $B_{22}$ 、 $B_{12}$ という順番でメモリ12からの読出しを行なう。

しかし、これを復号器13でそのまま再生した像 $R_1$ はオリジナル0を時計方向に90度回転した像にはなっていない。これは、ブロック位置は確かに回転されているが、各ブロックの内部がそのままのためである。従つて、ブロック内部を同じように時計方向に90度回転させて、初めて完全な再生画像 $R_2$ を得ることができる。このために、復号器13で復号化されたデータを回転器14でそれぞれの処理内容に応じて処理し、各ブロック内の画素の並び換えをしなければならない。

即ち、第7図に示す如くブロックサイズが

なされたもので、画像の回転や対称処理に適したデジタル画像の符号化方式を提供することを目的とする。詳しくは、本発明では、ブロックの符号をブロックのパターンを表わすコードと、そのパターンの回転や対称を表わすコードで構成する。こうすることで、ブロック内の回転や鏡像の処理を符号の回転、対称のモードを表わすコードに処理を施すことで、符号そのものを回転したパターンの符号に変えるものである。

#### 〔実施例〕

以下、本発明を好ましい実施例に基づいて説明する。本発明の概念を説明したのが、第1図である。

オリジナル画像0は4×4のブロックに分割されて、それぞれのブロックは符号器1によつて、そのブロックのパターン構造を表すパターンコードとそのパターンの回転・対称の状態を表す状態コードとを有するブロックコードに符号化される。このブロックコードがメモリ2に

4×4であり、ブロックを構成する画素位置を $x(1, j)$ で示すとすると、ブロックを構成する各画素に対して、回転器14において以下の如くの位置変換処理を必要としていた。

$x(1, j) \rightarrow x(j, 5-1)$  90度回転  
 $\rightarrow x(5-1, 5-j)$  180度回転  
 $\rightarrow x(5-j, 1)$  270度回転  
 $\rightarrow x(1, 5-j)$  鏡像  
 $\rightarrow x(j, 1)$  90度回転+鏡像  
 $\rightarrow x(5-1, j)$  180度回転+鏡像  
 $\rightarrow x(5-j, 5-1)$  270度回転+鏡像

回転方向は時計方向、鏡像は1軸に対してとる。

しかし、このような処理をハードウェアでしかもリアルタイムに実行することは、困難である。また、画像の処理内容に従つて、ブロック内の画素の処理の仕方も変えなければならないので、この問題は大きくなる。

#### 〔目的〕

本発明は、このような問題を解決するために

記憶される。

そして、ここでも画像を時計方向に90度回転するとは、回転処理を要しない場合の読出し順 $B_{11}$ 、 $B_{12}$ 、 $B_{21}$ 、 $B_{22}$ に対して、メモリ2から各ブロックコードをこの処理のために、 $B_{21}$ 、 $B_{11}$ 、 $B_{22}$ 、 $B_{12}$ の順番で読み出すとともに、回転器5で各ブロックコードの回転モードを表す状態コードに処理を施して、ブロックコードそのものをオリジナルのブロックのパターンを90度回転したパターンのブロックコードへと変換する。

このようにして、回転処理されたブロックコードを復号器5で復号して像を再生すれば、オリジナル0の90度回転像 $R$ を得ることができ、ブロック内の各画素に対する個別の処理は不要となる。

ところで、デジタル画像のブロック符号化方式の中で、ブロックのパターンを符号化する方式の代表的なものとして、ベクトル量子化による符号化方式が良く知られている。このベク

トル量子化はブロックの画素をベクトルの要素として考えて $4 \times 4$ のブロックなら、入力画像のブロックを16次元ベクトルとしてとらえる。そして、同じ16次元の空間でいくつかのベクトルを再生ベクトルとして登録して、入力ベクトルを画素が最小となるような再生ベクトルへ写像する。たとえば、入力画像が1又は0しかとらない2値画像とすると $4 \times 4$ のブロックには $2^{16}$ 通りのパターンが存在するが、異なるいくつかのパターンに同一コードを割付けることにより再生像のパターンとして例えば $2^8$ 通りのパターンを登録すれば情報量としては、16ビットから8ビットへと減らすことができる。尚、再生画像の画素を更に許容できる場合には $2^{16}$ 通りのパターンを $2^8$ 通りのパターンに圧縮してもよい。

このベクトル量子化に際し、符号化の際にはブロックのパターンは再生ベクトルを示すアドレスとして符号化される。またハードウェアとして実現する際には符号化或いは復号化のため

の変換テーブルを記憶したメモリROM等により写像を比較的容易に実現することができる。このようにベクトル量子化による符号化は、ブロックのパターン構造を符号化する方式ととらえることができる。

そして、前述の様にこのような符号化において、パターンの構造を表すパターンコードと、そのパターンの回転・対称の状態を表す状態コードとによりブロックのパターンを表現して符号化するものである。これは、再生パターンとして登録する際に回転・対称処理により同一パターンとなる複数のパターンのうちの1つのパターンを基本パターンとして考え、この基本パターンを回転・対称処理して得ることのできる複数のパターンを登録し、これら基本パターンに派生する複数パターンを表すパターンコードは基本パターンのものと共通とし、その回転・対称モードを表す状態コードを各パターンを表すパターンコードと合わせる。これにより、状態コードのみを変更することにより、例えば所

望の角度回転したパターンのコードを得ることができる。

第2図は、以上の概念に基づく符号化の具体例であつて、 $4 \times 4$ のブロックの画像を8ビットで符号化する場合を示し、8ビットのブロックコードのうち上位5ビットに画像パターンを示すパターンコードPCを、また、下位3ビットに回転・対称のモードを表わす状態コードBCを割り当てる。尚、この例では $2^{16}$ 通りのパターンを $2^8$ 通りのパターンコードに圧縮している。

第3図は第2図示のブロックコードにより符号化されている画像BPを時計方向に90度回転する動作例を示すもので、画像BPに対応したパターンコードPCを「1111」とし、また、画像BPを基本パターンとして登録してあるとする。前述の如く、基本パターンは状態コードBCとして“000”とすれば、基本パターンのブロックコードは“11111000”となる。これを90度回転させたパターンについては、状態コードBCとして、時計方向への90度回転

を“001”とすればこの回転処理後のパターンBP'を表わすブロックコードは、“11111001”となる。

この時この回転・対称モードを表すコード付けを以下の様に行なうことで、状態コードBC<sup>C</sup>そのものを再生ベクトルとして利用することができる。そのためには、状態コードBC<sup>C</sup>を基本パターンに対する絶対的な回転・対称の状態を表すだけのコードとしてとらえるのではなく、相対的な回転・対称の状態を表すコードとする。こうすることで、もとの状態コードBCが基本パターンに対してどのような状態にあつても同一の処理とすることで処理前と処理後では相対的に同一の状態を作り出すことができる。

以下に状態コードBCの例を示す。また、第4図に状態コードBCによる画像の状態遷移図を示す。

基本パターン	000
90度回転	001
180度回転	010

270度回転	011
鏡像	111
90度回転+鏡像	110
180度回転+鏡像	101
270度回転+鏡像	100

即ち、基本パターンを示す状態コード80を000とし、8つの状態を上述の様に3ビットのコードで表わす。そして、第4図の状態遷移図に示すように、8つの回転対称の状態間に規則性をもたせてある。

すなわち、3ビットの状態コード80のうち下位2ビットに1を加えることで、そのパターンを右(時計)方向に90度回転させたパターンのコードにすることができる。また、逆に下位2ビットから1を引く毎に左(反時計)方向にパターンを90度回転させることができる。さらに、状態コード80の3ビットの各ビットを反転させる(0なら1に、1なら0にする)ことで、そのパターンの鏡像とすることができる。

示する。

$R_1$	$R_0$	処 理 内 容
0	1	時計方向に90度回転(反時計方向に270°)
1	0	180度回転
1	1	時計方向に270度回転(反時計方向に90°)
0	0	処理なし

21は2ビットの加算器で、入力する状態コード80のうちの下位2ビット( $C_1, C_0$ )を前述の信号 $R_1, R_0$ により回転処理するもので、具体的には $C_1 C_0 + R_1 R_0$ の加算動作し、加算結果の下位2ビットを $Y_1, Y_0$ として出力する。

22~25は入力信号のレベルを反転するインバータ、26~31はアンドゲート、32~34はオアゲートである。状態コード80の上位1ビットの $C_2$ 及び加算器21の2つの出力 $Y_1, Y_0$ は夫々インバータ22~24により反転されてアンドゲート26, 28, 30に入力され、また、 $C_2, Y_1, Y_0$ はそのままアンドゲート27, 29, 31に入力される。アンドゲート26, 28, 30には信

号 $M$ が入力され、また、アンドゲート27, 29, 31には信号 $M$ をインバータ25で反転した信号が入力される。

このようなコード化をすれば、ベクトル量子化のような符号化の際に、入力ブロックを再生ベクトルへ写像するのに基本的パターンとか、それを何度回転させたパターンとかいようなことを意識せずにすべて同一レベルの再生ベクトルとして扱うことができる。尚、この時注意しなければならないことは、ある基本パターンを回転・対称処理して得られるパターンを再び基本パターンとしないことである。

第5図は、状態コード80の処理を実現するための第1図示の回転器5の詳細な構成例である。 $C_2, C_1, C_0$ は第3図で示した回転・対称モードを表す3ビットの状態コード80である。

また、 $R_1, R_0$ はパターンの回転角、また、 $M$ は鏡像処理の有無を示す信号であつて、例えば、キーボードやデジタイザ等の入力装置を用いたオペレータからの画像処理指令に従った信号である。信号 $M$ が"1"で状態コード80の3ビットの反転、即ち、鏡像処理を指示する。また、信号 $R_1, R_0$ の組合せにより以下の様な処理を指

示する。

従つて、信号 $M$ が"0"の場合はアンドゲート27, 29, 31の入力、即ち、 $C_2, Y_1, Y_0$ が夫々オアゲート32, 33, 34を通して $C_2, C_1, C_0$ として出力される。また、信号 $M$ が"1"の場合はアンドゲート26, 28, 30の入力、即ち、 $C_2, Y_1, Y_0$ を反転した値が夫々オアゲート32, 33, 34を通して $C_2, C_1, C_0$ として出力される。この様にして、加算器21にて回転処理された状態コード80が鏡像処理される。

以上の如く、第5図示の回転器5により状態コード80が処理指令に従つて、変換される。尚、ブロックコードのうちパターンコードは回転器5では何ら処理されない。

この様にして処理された状態コード80及びパターンコードPCを含むブロックコードは第1図示の復号器において、前述した如くの変換テーブルの検索技術を用いて画像パターンに復

号される。

以上説明した実施例では符号化のブロックを  $4 \times 4$  としたが、これに限るものではなく、また、符号化すべき画素も白/黒の2値に限らず、階調性をもつものであつてもよい。また、パターンコードは5ビット、状態コードは3ビットに夫々限らず、圧縮率や処理内容に応じてビット数は増減するものである。

〔効果〕

以上説明したように、ベクトル量子化のようなブロックのパターンを符号化する際に、ブロックの符号をパターンを換わすコードとそのパターンの回転および対称の状態等を表すコードとから構成することで、符号化した状態で回転・対称のような処理をする際にブロック内の画素の処理も符号に簡単な操作を施すことで高速に実現することができる。

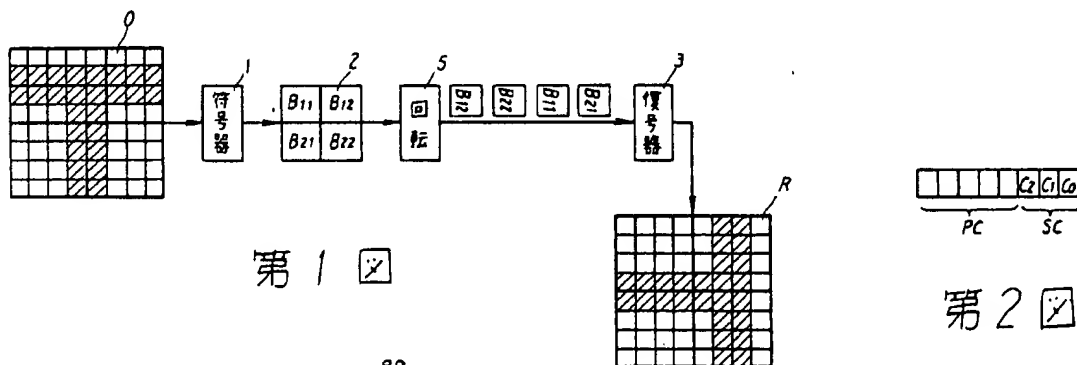
また、ベクトル量子化のように、その再生像に歪を生じる符号化方式においては、その回転・対称のブロックは必ず存在するから、どのよう

な回転・対称をしても、その再生像の間では画質の差が生じず、たとえば、 $90^\circ$ 回転した像も、 $180^\circ$ 回転した像も画質としては全く同一のものを得ることが出来ることは、実用上大きなメリットである。

#### 4. 図面の簡単な説明

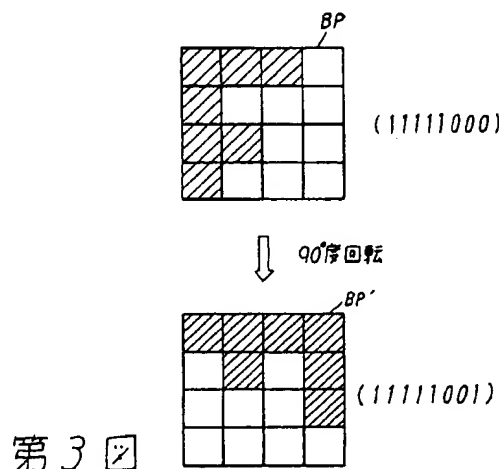
第1図は本発明による符号化の概念を示す図、第2図は本発明による符号化に用いるコードの例を示す図、第3図は画像処理の一例を示す図、第4図は状態遷移図、第5図は回転器の構成例を示す図、第6図は従来の符号化を示す図、第7図は従来の画像処理の一例を示す図であり、1は符号器、5は回転器、3は復号器である。

出願人 キヤノン株式会社  
代理人 丸 島 儀 一

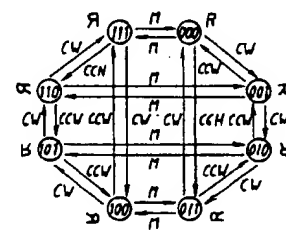


第1図

第2図



第3図



第4図



PAT-NO: JP362140554A  
DOCUMENT-IDENTIFIER: JP 62140554 A  
TITLE: ENCODING SYSTEM FOR DIGITAL IMAGE  
PUBN-DATE: June 24, 1987

INVENTOR-INFORMATION:

NAME  
SUZUKI, YOSHIYUKI  
FUNADA, MASAHIRO  
SATO, MAMORU  
SATO, YUKIO

ASSIGNEE-INFORMATION:

NAME	COUNTRY
CANON INC	N/A

APPL-NO: JP60281641

APPL-DATE: December 13, 1985

INT-CL (IPC): H04N001/415

ABSTRACT:

PURPOSE: To process a picture element in a block at a high speed by performing a simple operation to a code, by constituting a code of the block, of a code for showing a pattern and a code for showing a rotation of its pattern and a symmetrical state, etc.

CONSTITUTION: When registering as a reproducing pattern, one pattern of plural patterns which become the same pattern by rotation/symmetry processings is considered to be a fundamental pattern, and a state code is aligned with a

pattern code for showing each pattern. In case of encoding an image of a block of  $4 \times 4$  by 8 bits, a pattern code PC for showing an image pattern, and a state code SC for showing a mode of a rotation and a symmetry are allocated to the upper 5 bits and the lower 3 bits, respectively. When the pattern code PC is denoted as '11111', and the fundamental pattern is set as a state code '000', a block code of the fundamental pattern becomes '11111000', and when a clockwise  $90^\circ$  rotation is denoted as '001', a block code after this rotation processing becomes '11111001'.

COPYRIGHT: (C)1987, JPO&Japio